拟合矩阵与两阶融合迭代加速推荐算法 *

王　帅，孙福振†，王绍卿，张　进，方　春

(山东理工大学 计算机科学与技术学院，山东 淄博 255049)

摘　要：传统的矩阵分解模型无法充分探索用户与物品在均值、偏置和特征之间的内在联系，提出拟合矩阵模型，通过构建用户与物品矩阵分别代表用户与物品特性，提高预测性能。矩阵分解模型在推荐系统领域有精度优势，但求解模型参数最常用的梯度下降法收敛速度缓慢。针对梯度下降法的上述缺陷，考虑与拟牛顿法进行融合，使得收敛速度加快。提出的算法命名为拟合矩阵与两阶融合迭代加速推荐算法(fitting matrix and two orders fusion iterative，FAST)。实验表明，FAST 算法比传统的非负矩阵分解(NMF)，奇异值矩阵分解(SVD)，正则化奇异值矩阵分解(RSVD）在平均绝对误差(MAE）与均方根误差(RMSE）上有下降，在迭代效率上有显著提高，缓解了精度与迭代效率难以平衡的问题。

关键词：拟合矩阵；矩阵分解；拟牛顿法；梯度下降；融合

中图分类号：TP301.6　　doi: 10.19734/j.issn.1001-3695.2018.07.0533

Accelerating recommendation algorithm using fitting matrix and two orders Fusion iterative

Wang Shuai, Sun Fuzhen, Wang Shaoqing, Zhang Jin, Fang Chun

(*College of Computer Science & Technology*，*Shandong University of Technology*, *Zibo Shandong* 255049, *China*)

Abstract: The traditional matrix decomposition model can not fully explored the intrinsic relationship between the user and the object in the mean, bias and characteristics. This paper proposed a fitting matrix model to improve the prediction performance by constructing the user and the item matrix to represent the characteristics of the user and the item respectively. The matrix decomposition model had the advantage of accuracy in the field of recommender system, but the gradient descent method, which was the most popular method to train parameters of model, had a slow convergence speed. Considering the above defects, this paper considered to accelerate the convergence speed using the convergence of quasi Newton method. We named the proposed algorithm as fitting matrix and two orders fusion iterative (FAST) algorithm. The experimental results showed that the FAST algorithm was better than the traditional non negative matrix decomposition (NMF) , singular value matrix decomposition (SVD) , and the regularized singular value matrix decomposition (RSVD) . FAST algorithm had a decrease with regard to the mean absolute error (MAE) and the root mean square error (RMSE) , and had a significant improvement in the iterative efficiency, which alleviated the problem that the accuracy was difficult to balance with the efficiency of the iteration.

Key words: fitting matrix; matrix decomposition; quasi Newton method; gradient descent; Fusion

0　引言

协同过滤推荐技术是推荐系统中应用最早和最为成功的技术之一。奇异值矩阵分解（SVD）是用于协同过滤的常用算法之一。而直接应用传统的奇异值矩阵分解算法协同过滤可能导致性能不佳。因而需要结合推荐系统数据稀疏性的特征提出行之有效的奇异值矩阵分解方法[1]。其中最著名的就是正则化奇异值矩阵分解（RSVD），在原有奇异值矩阵分解的基础上加入正则化因子和偏置因子，利用线性回归方法让正则化因子与偏置因子产生双向交互作用，提高奇异值矩阵分解的预测能力[2]。不同于奇异值矩阵分解（SVD），协同过滤中还有一种矩阵分解的方式为非负矩阵分解（NMF），这种方法也是协同过滤推荐系统中一种比较流行的算法之一[3]。

而文献[4]提出将梯度下降与拟牛顿法结合，采用一种随机正则化（RES）的拟牛顿法用于求解随机函数的凸优化问题，对拟牛顿法在高维矩阵中的发散问题与海森矩阵的逆矩阵的计算与存储问题提出一种有效的改进思路。然而这种结合方式的预测精度介于梯度下降与拟牛顿法之间，以降低部分预测精度换取二者的有效结合。文献[5]也提出了一种将拟牛顿法与其他方法结合的思路，称为混合人工蜂群策略。通过统计分析，所提出的训练策略相比于用拟牛顿法训练的 DNN 分类器具有更好的分类性能。这种方法适合于数据稀疏性低且数据特征明显的数据（如图像识别）。不适用于数据稀疏性很高的推荐系统领域。

文献[6~8]中的模型都借鉴了传统协同过滤技术中的基于用户推荐模型，而这类模型最大的优势是不需要迭代数据，从而时间效率较高，但是与具有迭代数据的模型（例如奇异值矩阵分解）相比推荐精度稍差。文献[910]都是近年来流行的基于排序学习模型，最终形成 top$N$ 推荐列表的文章，这类模型优势在于算法复杂度低，不需要人为设定超参数。但是精度相比协同过滤中具有迭代数据的模型略低，而且难以进

收稿日期：2018-07-30；**修回日期：**2018-09-28　**基金项目：**国家自然科学基金资助项目 02280)；山东省自然科学基金资助项目（ZR2014FQ028）

作者简介：王帅（1993-），男，硕士研究生，主要研究方向为智能信息计算与处理、推荐系统；孙福振（1978-），男（通信作者），副教授，硕导，博士，主要研究方向为数据挖掘、智能信息处理（sunfuzhen@sdut.edu.cn）；王绍卿（1981-），男，博士，主要研究方向为数据挖掘、推荐系统；方春（1981-），女，博士，主要研究方向为数据挖掘、模式识别；张进（1995-），男，硕士研究生，主要研究方向为智能信息处理.

行扩展。

综上所述，为提升奇异值矩阵分解的训练效率，以及提高预测精度，本文提出了一种基于拟合矩阵和两阶融合迭代加速推荐算法。

1　相关概念

1.1　SVD 在推荐系统中的应用

奇异值分解是将一个 $m \times n$ 的矩阵 A，通过公式 $A = U\Sigma V^T$ 分解为三个矩阵。其中一个 $m \times m$ 的方阵，Σ 是一个 $m \times n$ 的奇异值矩阵，V 是一个 $n \times n$ 的方阵。U 与 V 的元素均是正交向量。

由于奇异值分解中的矩阵 A 是一个完整矩阵，而推荐系统的评分矩阵 R 是一个不完整矩阵并且稀疏。导致推荐系统中应用 SVD 时，不能直接使用公式 $R = U\Sigma V^T$。假设用户与物品之间没有直接联系，人为定义特征维度 $r = f$。用户与物品通过与 f 的联系将评分矩阵 R 分解成为两个特征维度矩阵 U 与 V，U 代表用户特征，是一个 $r \times m$ 的矩阵，m 代表用户数目。V 代表物品特征，是一个 $r \times n$ 的矩阵，n 代表物品数目。最终通过计算 U 与 V，使得 $U^T V$ 的结果越接近 R 越好。

为了 $U^T V$ 的结果不断接近 R，梯度下降法是最常用的求解方法。梯度下降法的基本原理就是使损失函数沿梯度下降的方向通过步步迭代接近极小值。迭代公式如下：

$$\theta_i = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1 \ldots, \theta_n) \tag{1}$$

其中：$-\frac{\partial}{\partial \theta_i}$ 代表负梯度方向，α 代表梯度方向的搜索步长，θ 代表参数进行迭代计算。

1.2　拟牛顿法原理

牛顿法的基本思想是：在现有的极小点估计值的附近对目标函数做二阶泰勒展开式，从而找到极小点的下一个估计值。假设当前的极小点估计值为 x_k。则目标函数在 x_k 附近的二阶泰勒展开式为

$$\varphi(x) = f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T * \nabla^2 f(x_k) * (x - x_k) \tag{2}$$

其中 ∇f 为 f 的梯度向量，$\nabla^2 f$ 为 f 的海森矩阵。由于极值的必要条件是 $\nabla \varphi(x) = 0$，设 ∇f 为 G，$\nabla^2 f$ 为 H，则牛顿法的迭代公式为

$$x = x_k - H_k^{-1} * G_k\,. \tag{3}$$

牛顿法要求目标函数必须有连续的一阶和二阶偏导数且海森矩阵必须正定，而牛顿法的应用条件在实际情况往往是无法满足的。拟牛顿法是在牛顿法的基础上进行的改进。拟牛顿法的基本思想是利用一阶偏导数求解近似海森矩阵的正定对称阵。近似海森矩阵的条件为

$$\nabla \varphi(x_k) \approx \nabla f(x_{k+1}) + H_{k+1} * (x_k - x_{k+1}) \to \begin{cases} 1. G_{k+1} - G_k \approx H_{k+1} * (x_{k+1} - x_k) \\ 2. x_{k+1} - x_k \approx H_{k+1}^{-1} * (G_{k+1} - G_k) \end{cases} \tag{4}$$

由上述两式作为拟牛顿法的近似约束条件，是拟牛顿法的核心。

2　算法结构分析

2.1　拟合矩阵模型

传统的推荐系统奇异值矩阵分解模型预测函数为

$$\sigma(p_i, q_j) = u + b_i + b_j + p_i^T q_j \tag{5}$$

其中 u 代表全部评分数据的均值，b_i 代表用户 i 的偏置量，b_j 代表物品 j 的偏置量，p_i 代表用户 i 的特征向量，q_j 代表物品 j 的特征向量。传统预测函数虽然将用户 i 与物品 j 分成两个特征矩阵的形式表示，但是特征值的初始化是随机的，而且在计算中不考虑用户与物品自身可能存在的对特征值的潜在影响。

本文引入的拟合矩阵模型将 SVD 预测公式中的整体均值量 u 替换为当前用户均值与当前物品均值的融合，这样更能够体现当前用户与当前物品的个性化表现。

首先定义用户 i 的均值矩阵 M_i 为 k 维，物品 j 的均值矩阵 M_j 为 k 维。初始化 M_i 与 M_j 由数据集计算得到标量 u_i 与 u_j 均分为 k 维矩阵。用户特征矩阵 S_i 与物品特征矩阵 Z_j 也是 k 维的矩阵,初始化随机生成。用户 i 的偏置矩阵 C_i 与物品 j 的偏置矩阵 C_j 同样也是 k 维的矩阵，初始化由标量 b_i 与 b_j 均分成 k 维矩阵。

为了让均值矩阵（M_i 与 M_j）和特征矩阵（S_i 与 Z_j）以及偏置矩阵（C_i 与 C_j）融合，引入拟合矩阵模型，为用户 i 与物品 j 分别构造用户矩阵 X_i 与物品矩阵 Y_j，以 k 维矩阵形式表示。用户矩阵 X_i 由用户 i 的均值矩阵 M_i、用户 i 的特征矩阵 S_i、用户 i 的偏置矩阵 C_i 的乘积形式见式（6）。

$$X_i = M_i \times (S_i \times C_i) \tag{6}$$

物品矩阵 Y_j 由物品 j 的均值矩阵 M_j 与物品 j 的特征矩阵 Z_j 和物品 j 的偏置矩阵 C_j 以乘积的形式表示为

$$Y_j = M_j \times (Z_j \times C_j) \tag{7}$$

由此每个用户或物品矩阵由均值矩阵（M_i 或 M_j），特征矩阵（S_i 或 Z_j）与偏置矩阵（C_i 或 C_j）三部分融合而成，构成双重拟合矩阵。式（6）和（7）的计算过程如图 1 所示。

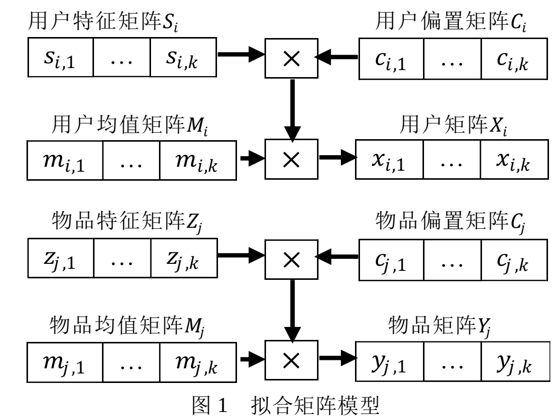


图 1　拟合矩阵模型

Fig. 1　Fitting matrix model

构造出的用户与物品的能够在一定程度上反映用户与物品在当前特征上的潜在影响。新的预测函数如式（8）所示。

$$\sigma(X_i, Y_j) = (M_i + M_j)/2 + b_i + b_j + X_i^T Y_j \tag{8}$$

2.2　两阶融合迭代加速模型

利用梯度下降作为一阶计算的处理方式，而拟牛顿法作为二阶计算的处理方式，使用推荐系统中的误差函数作为目标函数，使得梯度下降与拟牛顿法并存，以两阶偏导数融合的形式进行计算。

对于求解 SVD 模型的参数，拟牛顿法并不能完全替代梯度下降法进行计算，原因主要有：拟牛顿法的学习效率很高，能够迅速拟合训练数据，易于导致过拟合；推荐系统误差函数中存在随机正则项。如，推荐系统误差函数为

$$\min_{b_i, b_j, X_i, Y_j} \sum_{(i,j)\in K} \left(R_{ij} - u - b_i - b_j - X_i^T Y_j\right)^2 + \delta\left(b_i^2 + X_i^2\right) + \delta\left(b_j^2 + Y_j^2\right) \tag{9}$$

基于以上原因，本文提出将梯度下降法与拟牛顿法融合进行计算的思路。对拟牛顿法的步长计算方式进行改进，利用拟牛顿法高效的收敛效率对预测结果与实际结果偏差较小的值进行收敛，而预测结果与实际结果偏差较高的值依然使用梯度下降进行收敛。拟牛顿法中步长 λ 的计算方式为一维搜索，非精确一维搜索由于存在超参数而梯度下降中已存在超参数 μ 与 δ，而算法中如果出现过多的超参数就会失去普适性的意义。若使用成熟的一阶偏导数或函数自身求解极小值则会因误差函数存在随机正则项从而使求解过程失去实际意义。

由此，对于步长 λ 的计算方式，通过观察误差函数的形式为完全的二次幂函数，因此利用二阶导数计算最终的步长 λ 成为可能。由数学定义可知，二阶导数定义是函数的拐点定义。而推荐系统误差函数可满足一般的二次函数特性，函数有且只有一个极小值点，并且极小值点也为拐点。同时在二阶导数的情况下，便完全将正则项转换为常数，从而能把正则项对极值的影响消除，再运用此极小值计算步长 λ。便能避开非精确一维线性搜索的超参数过多以及一阶导数或者利用函数自身求解极小值导致的随机正则项问题。

对拟牛顿法步长的改进处理方式：

a）对误差函数求关于 X_i 梯度并对 X_i 求关于 λ_i，M_i，S_i，C_i 梯度：

$$\nabla X_i=\sum_{i\in K}\delta X_i-\left[R_{ij}-\sigma\left(X_i,Y_j\right)\right]Y_j? \tag{11}$$

$$\nabla\lambda_i=\frac{\partial X_i}{\partial\lambda_i}=\sum_{i\in K}d_i^{\lambda}*M_i\times(S_i\times C_i)? \tag{12}$$

$$\nabla M_i=\frac{\partial X_i}{\partial M_i}=\sum_{i\in K}d_i^{M}*\lambda_i^{M}*(S_i\times C_i)? \tag{13}$$

$$\nabla S_i=\frac{\partial X_i}{\partial S_i}=\sum_{i\in K}d_i^{S}*\lambda_i^{S}*(M_i\times C_i)? \tag{14}$$

$$\nabla C_i=\frac{\partial X_i}{\partial C_i}=\sum_{i\in K}d_i^{C}*\lambda_i^{C}*(M_i\times S_i)? \tag{15}$$

b）对误差函数求关于 Y_j 梯度并对 Y_j 求关于 λ_j，M_j，Z_j，C_j 梯度：

$$\nabla Y_j=\sum_{j\in K}\delta Y_j-\left[R_{ij}-\sigma\left(X_i,Y_j\right)\right]X_i? \tag{16}$$

$$\nabla\lambda_j=\frac{\partial Y_j}{\partial\lambda_j}=\sum_{j\in K}d_j^{\lambda}*M_j\times\left(Z_j\times C_j\right)? \tag{17}$$

$$\nabla M_j=\frac{\partial Y_j}{\partial M_j}=\sum_{j\in K}d_j^{M}*\lambda_j^{M}*\left(Z_j\times C_j\right) \tag{18}$$

$$\nabla Z_j=\frac{\partial Y_j}{\partial Z_j}=\sum_{j\in K}d_j^{Z}*\lambda_j^{Z}*\left(M_j\times C_j\right) \tag{19}$$

$$\nabla C_j=\frac{\partial Y_j}{\partial C_j}=\sum_{j\in K}d_j^{C}*\lambda_j^{C}*\left(M_j\times Z_j\right) \tag{20}$$

c）将 d_i 与 λ_i 代入 X_i，将 d_j与λ_j代入Y_j，并计算误差 *loss* 关于 λ_i 与 λ_j 二阶偏导：

$$\sum_{i\in K}X_i=X_i+\lambda_i d_i^{\lambda};\sum_{j\in K}Y_j=Y_j+\lambda_j d_j^{\lambda}$$

$$\sum_{i,j\in K}\frac{\partial loss}{\partial X_i}\frac{\partial X_i}{\partial\lambda_i}+\frac{\partial loss}{\partial Y_j}\frac{\partial Y_j}{\partial\lambda_j}=0? \tag{21}$$

d）将 d_i^M 与 λ_i^M 代入 X_i，将 d_j^U与λ_j^U代入Y_j，并计算误差 *loss* 关于 M_i 与 M_j 二阶偏导

$$\sum_{i\in K}X_i=X_i+\lambda_i^M d_i^M;\sum_{j\in K}Y_j=Y_j+\lambda_j^M d_j^M$$

$$\sum_{i,j\in K}\frac{\partial loss}{\partial X_i}\frac{\partial X_i}{\partial M_i}+\frac{\partial loss}{\partial Y_j}\frac{\partial Y_j}{\partial M_j}=0? \tag{22}$$

e）将 d_i^S 与 λ_i^S 代入 X_i，将 d_j^Z与λ_j^Z代入Y_j，并计算误差 *loss* 关于 S_i 与 Z_j 二阶偏导

$$\sum_{i\in K}X_i=X_i+\lambda_i^S d_i^S;\sum_{j\in K}Y_j=Y_j+\lambda_j^Z d_j^Z$$

$$\sum_{i,j\in K}\frac{\partial loss}{\partial X_i}\frac{\partial X_i}{\partial S_i}+\frac{\partial loss}{\partial Y_j}\frac{\partial Y_j}{\partial Z_j}=0 \tag{23}$$

f）将 d_i^C 与 λ_i^C 代入 X_i，将 d_j^C与λ_j^C代入Y_j，并计算误差 *loss* 关于 C_i 与 C_j 二阶偏导

$$\sum_{i\in K}X_i=X_i+\lambda_i^C d_i^C\quad\sum_{j\in K}Y_j=Y_j+\lambda_j^C d_j^C$$

$$\sum_{i,j\in K}\frac{\partial loss}{\partial X_i}\frac{\partial X_i}{\partial C_i}+\frac{\partial loss}{\partial Y_j}\frac{\partial Y_j}{\partial C_j}=0 \tag{24}$$

运用第 a）和 b）步求出需要的代数式，再利用 c）~f）的二阶公式求出对应的 λ_i，λ_i^M，λ_i^S，λ_i^C 和 λ_j，λ_j^M，λ_j^Z，λ_j^C。取二者绝对值最小者为步长 λ_{ij}，λ_{ij}^M，λ_{ij}^{SZ}，λ_{ij}^C。为了控制拟牛顿法可能出现的无法保证函数稳定下降的问题。使用 λ^2 控制 λ 的大小范围，一般在 0～1 内为有效值。

2.3　两阶融合方式

经过改进拟牛顿法的步长计算方式之后，利用计算得出的步长作为两阶融合方式的判断标准。下次迭代时，如果步长计算数值不属于 0～1 范围内，则此次迭代使用梯度下降。如果步长计算数值属于 0～1 范围内，则此次迭代使用拟牛顿法。这样的融合方式加快了收敛速度。经过实验验证，步长的阈值为 0～1 时，算法拥有最好的收敛速度与精度。由于梯度下降迭代步长是人为设置的超参数，不具有计算参考意义，因此无法使用梯度下降的步长作为两阶融合方式的判断标准。算法符号表如表 1 所示，算法流程图如图 2 所示。

表 1　算法符号表

Table 1　Algorithm symbol table

拟合矩阵符号表		两阶融合符号表	
u	均值(标量)	δ	随机正则项参数
b_i	用户偏置(标量)	∇X_i	用户矩阵梯度
b_j	物品偏置(标量)	$\nabla\lambda_i$	用户步长 λ 梯度
p_i	用户特征向量	∇M_i	用户均值梯度
q_j	物品特征向量	∇S_i	用户特征梯度
M_i	用户均值矩阵	∇C_i	用户偏置梯度
S_i	用户特征矩阵	d_i	用户搜索方向
C_i	用户偏置矩阵	∇Y_j	物品矩阵梯度
X_i	用户矩阵	$\nabla\lambda_j$	物品步长 λ 梯度
M_j	物品均值矩阵	∇M_j	物品均值梯度
Z_j	物品特征矩阵	∇Z_j	物品特征梯度
C_j	物品偏置矩阵	∇C_j	物品偏置梯度
Y_j	物品矩阵	d_j	物品搜索方向

3　算法伪代码分析

拟合矩阵与两阶融合迭代加速推荐算法基于拟合矩阵算法与两阶融合迭代加速算法两个主要部分组成。其中，两阶融合迭代加速算法分为四部分，目的是对不同类型的矩阵计算不同的迭代步长，迭代步长分开计算的原因是不同的矩阵代表不同的信息，组成的方式也不同。主要包括对用户与物

品矩阵计算两阶融合的迭代步长；对用户与物品均值矩阵计算迭代步长；对用户与物品特征矩阵计算迭代步长；对用户与物品偏置矩阵计算迭代步长。以下是具体算法的伪代码表示：

算法 1　拟合矩阵算法伪代码

$PreRate \leftarrow 0.5 * M_i + 0.5 * M_j + b_i + b_j$

for *feature* $\leftarrow$ 0 *to Count* ***do***

$X_i \leftarrow M_i \times (S_i \times C_i)$

$Y_i \leftarrow M_j \times (Z_j \times C_j)$

$\sigma(X_i, Z_j) \leftarrow PreRate + X_i^T Y_j$

End for

初始化均值矩阵（M_i与M_j）、特征矩阵（S_i与Z_j）与偏置矩阵（C_i与C_j）

↓

利用初始矩阵拟合计算得到用户与物品的初始化矩阵X_i与Y_j

↓

1. 初始化用户与物品近似海森矩阵B_Y^X.
2. 牛顿搜索方向：$d_Y^X = -B_Y^{X^{-1}} \cdot g_Y^X$.
3. 利用两阶融合步长计算方式计算λ_{ij}，判断λ_{ij}是否属于阈值范围内。
3.1 若属于则继续拟牛顿法的计算步骤：$s_Y^X = \lambda_{ij} d_Y^X$；$x' = x + s_Y^X$；$f_Y^X = g' - g$.
3.2 利用s_Y^X与f_Y^X更新近似海森矩阵：$B_Y^{X'} = B_Y^X + f_Y^X f_Y^{X^T} / f_Y^{X^T} s_Y^X - B_Y^X s_Y^X s_Y^{X^T} B_Y^X / s_Y^{X^T} B_Y^X s_Y^X$.
4. 若不属于则利用梯度下降法$X_i \leftarrow X_i + \nabla X_i'$，$Y_j \leftarrow Y_j + \nabla Y_j'$更新用户与物品矩阵。

↓

1. 初始化均值近似海森矩阵B_M. 2. 牛顿搜索方向：$d_M = -B_M^{-1} \cdot g_M$. 3. 利用二阶偏导数求解均值矩阵的步长$\lambda_{ij}^M$. 4. 利用 3.1 和 3.2 的公式更新$B_M$.	1. 初始化特征近似海森矩阵B_{SZ}. 2. 牛顿搜索方向：$d_{SZ} = -B_{SZ}^{-1} \cdot g_{SZ}$. 3. 利用二阶偏导数求解均值矩阵的步长$\lambda_{ij}^{SZ}$. 4. 利用 3.1 和 3.2 的公式更新$B_{SZ}$.	1. 初始化偏置近似海森矩阵B_C. 2. 牛顿搜索方向：$d_C = -B_C^{-1} \cdot g_C$. 3. 利用二阶偏导数求解均值矩阵的步长$\lambda_{ij}^C$. 4. 利用 3.1 和 3.2 的公式更新$B_C$.

↓

用户与物品矩阵通过二阶偏导数参与均值矩阵、特征矩阵与偏置矩阵的更新计算。用户与物品矩阵与其子矩阵的更新通过并行化计算。

↓

拟合矩阵模型预测评分

图 2　算法流程图

Fig. 2　Algorithm flow chart

算法 1 表示拟合矩阵计算过程。输入为 M_i，S_i，Z_j，C_i，C_j，b_i，b_j。通过矩阵内对应向量元素乘积构成用户与物品矩阵对应向量元素，输出用户与物品矩阵的向量元素点积。最终与均值和偏置值组成预测值。

算法 2　两阶融合迭代加速模型 λ_{ij}（用户与物品矩阵两阶融合迭代步长）伪代码：

$\nabla X_i \leftarrow \mu \nabla X_i$

$\nabla Y_i \leftarrow \mu \nabla Y_i$

$X_i' \leftarrow X_i + \lambda_i d_i$

$Y_i' \leftarrow Y_i + \lambda_j d_j$

$\lambda_i \; or \; \lambda_j \leftarrow \nabla X_i * X_i' + \nabla Y_i * Y_i'$

$\lambda_{ij} \leftarrow \min_{\lambda \in R}(\lambda_i \text{ or } \lambda_j)$

if $(\lambda$ *is dissatisfy the condition*$)$ ***then***

$X_i \leftarrow X_i + \nabla X_i$

$Y_i \leftarrow Y_i + \nabla Y_i$

End if

$\lambda_{ij} \leftarrow \lambda_{ij}^2$

算法 2 表示两阶融合迭代加速关于用户与物品矩阵的步长融合计算过程。输入为 X_i 和 Y_i， $\mu \nabla X_i$ 和 $\mu \nabla Y_i$ 通过求解梯度下降学习率与其自身梯度乘积而得。将 d_i 和 d_j（d_i 与 d_j 由其对应的负近似海森矩阵的逆矩阵与 $\mu \nabla X_i$ 和 $\mu \nabla Y_i$ 的乘积而得）与 λ_i 和 λ_j 作为参数加入 X_i 和 Y_i。通过二阶偏导 $\nabla X_i * X_i' + \nabla Y_i * Y_i'$ 求解 λ_i 和 λ_j。对应取 λ_i 和 λ_j 中绝对值最小为 λ_{ij}，如果不满足阈值范围，则用户与物品矩阵通过梯度下降进行计算。如果满足则 λ_{ij}^2 作为拟牛顿法的关于用户与物品矩阵的搜索步长因子。

算法 3　两阶融合迭代加速模型 λ_{ij}^M（用户与物品均值矩阵迭代步长）伪代码：

$\nabla M_i \leftarrow \mu \nabla M_i$

$\nabla M_j \leftarrow \mu \nabla M_j$

$M_i' \leftarrow M_i + \lambda_i^M d_i^M$

$M_j' \leftarrow M_j + \lambda_j^M d_j^M$

$\lambda_i^M \; or \; \lambda_j^M \leftarrow \nabla M_i * M_i' + ?\nabla M_j * M_j'$

$\lambda_{ij}^M \leftarrow \min_{\lambda \in R}(\lambda_i^M \; ?or \; \lambda_j^M)$

$\lambda_{ij}^M \leftarrow \lambda_{ij}^{M2}$

算法 4　两阶融合迭代加速模型 λ_{ij}^{SZ}（用户与物品特征矩阵迭代步长）伪代码：

$\nabla S_i \leftarrow \mu \nabla S_i$

$\nabla Z_j \leftarrow \mu \nabla Z_j$

$S_i' \leftarrow S_i + \lambda_i^S d_i^S$

$Z_j' \leftarrow Z_j + \lambda_j^Z d_j^Z$

$\lambda_i^S \; or \; \lambda_j^Z \leftarrow \nabla S_i * S_i' + \nabla Z_j ?* Z_j'$

$\lambda_{ij}^{SZ} \leftarrow \min_{\lambda \in R}(\lambda_i^S \; ?or \; \lambda_j^Z)$

$\lambda_{ij}^{SZ} \leftarrow \lambda_{ij}^{SZ2}$

算法 5　两阶融合迭代加速模型 λ_{ij}^C（用户与物品偏置矩阵迭代步长）伪代码：

$\nabla C_i \leftarrow \mu \nabla C_i$

$$\nabla C_j \leftarrow \mu \nabla C_j$$

$$C_i^{'} \leftarrow C_i + \lambda_i^{C} d_i^{C}$$

$$C_j^{'} \leftarrow C_j + \lambda_j^{C} d_j^{C}$$

$$\lambda_i^{C}\ or\ \lambda_j^{C} \leftarrow \nabla C_i * C_i^{'} + ?\nabla C_j * C_j^{'}$$

$$\lambda_{ij}^{C} \leftarrow \min_{\lambda \in R}\left(\lambda_i^{C}\ ?or\ \lambda_j^{C}\right)$$

$$\lambda_{ij}^{C} \leftarrow \lambda_{ij}^{C2}$$

算法 3~5 表示两阶融合迭代加速关于用户与物品均值矩阵，特征矩阵与偏置矩阵的步长计算过程。输入为 M 、SZ 、C 。∇M 、∇SZ 、∇C 通过求解梯度下降学习率与其自身梯度乘积而得。将 d_i 和 d_j （ d_i 和 d_j 由其对应的负近似海森矩阵的逆矩阵与其梯度的乘积而得）与 $\lambda_i^{'}$ 和 $\lambda_j^{'}$ 作为参数加入 M 、SZ 、C 。通过二阶偏导 $\nabla *' + ?\nabla *'$ 求解对应的 $\lambda_i^{'}$ 和 $\lambda_j^{'}$ 。对应取 $\lambda_i^{'}$ 和 $\lambda_j^{'}$ 中绝对值最小为 $\lambda_{ij}^{'}$ ，$\lambda_{ij}^{'2}$ 最终作为拟牛顿法的关于用户与物品均值矩阵，特征矩阵与偏置矩阵的搜索步长因子。不同于用户与物品矩阵的融合计算过程，由于 M 、SZ 、C 的目标函数没有随机化正则项，因此计算 M 、 SZ 、 C 的步长可以不需要与梯度下降融合。

4　实验数据分析

4.1　实验数据集

实验所用的数据集是美国 Minnesota 大学 GroupLens 小组开发的 MovieLens 站点所提供的数据集（https://grouplens.org/datasets/movielens/）与电影租赁网址 Netflix 的数据集。本文共使用三个不同大小的数据集，分别为： 6040 个用户对 3900 部电影的 100 万条评分的数据；71567 个用户对 10681 部电影给出的 1000 万条评分的数据；480189 个匿名用户对大约 17770 部电影作的大约 1 亿次评分的数据。将实验数据集划分为训练集和测试集，其中训练集占 80%，测试集占 20%。

4.2　评价标准

本文采用的推荐质量评价标准是平均绝对误差（mean absolute error，MAE）与均方根误差（root mean squared error，RMSE）。公式如下：$MAE = \frac{\sum_{i,j \in T} \left| r_{ij} - \hat{r}(i,j) \right|}{|T|}$ ，

$$RMSE = \sqrt{\frac{\sum_{i,j \in T} \left(r_{ij} - \hat{r}(i,j) \right)^2}{T}}$$ 。

4.3　实验对比

实验引入奇异值矩阵分解（SVD）[1]，随机正则化的奇异值矩阵分解（RSVD）[2]算法，非负矩阵分解（NMF）[3]与本文提出的拟合矩阵与两阶融合迭代加速（FAST）进行均方根误差以及迭代效率的比较。梯度下降迭代步长参数选取为 0.04

4.3.1 迭代次数与时间对比结果

图 3~5 分别是 1M、10M 和 Netflix 数据集的迭代次数对比图。从对比图中可以观察到，NMF、SVD 和 RSVD 模型使用梯度下降的情况下需要几十次才可以收敛。而 FAST 模型在第 8 次或第 9 次迭代就可以收敛，收敛的曲线在数据不断扩大的情况下越接近直线。实验表明 FAST 模型具有较高的迭代效率。

表 2　FAST 模型与其他模型的时间数据 /min

Table 2　Time data of FAST and other models /min

模型	NMF	SVD	RSVD	FAST
1M	10	15	18	12
10M	143	216	262	165

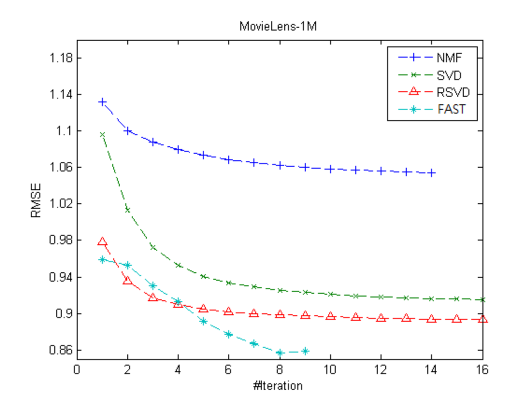


图 3　FAST 与其他模型在 MovieLens-1M 数据集中的迭代对比图

Fig. 3　Comparison of FAST and other models in MovieLens-1M dataset

图 4　FAST 与其他模型在 MovieLens-10M 数据集中的迭代对比图

Fig. 4　Comparison of FAST and other models in MovieLens-10M dataset

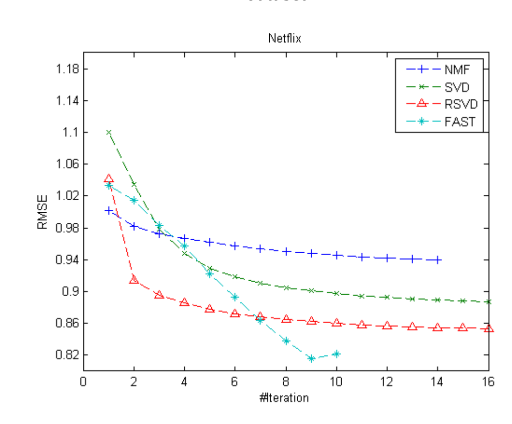


图 5　FAST 与其他模型在 Netflix 数据集中的迭代对比图

Fig. 5　Comparison of FAST and other models in Netflix dataset

4.3.2 实验 RMSE、MAE 对比结果

表 3 和 4 分别是 1M、10M，Netflix 数据集的 RMSE、MAE 结果对比。从表 3 和表 4 的数据可以看出，FAST 相比 NMF，SVD，RSVD 的均方根误差与平均绝对误差更小。且随着数据集的不断扩大，均方根误差与平均绝对误差减小的越明显，也就是预测效果越好。

表 3 FAST 模型与其他模型的 RMSE 数据

Table 3 The RMSE data of FAST and other models

模型	NMF	SVD	RSVD	FAST
1M	1.054	0.914	0.893	0.856
10M	0.996	0.872	0.857	0.765
Netflix	0.941	0.887	0.853	0.821

表 4 FAST 模型与其他模型的 MAE 数据

Table 4 The MAE data of FAST and other models

模型	NMF	SVD	RSVD	FAST
1M	0.810	0.723	0.711	0.670
10M	0.764	0.682	0.661	0.584
Netflix	0.755	0.665	0.639	0.614

5 结束语

个性化作为推荐系统的本质需求，通过关注用户与物品的特征构造特定的推荐方式已成为推荐系统的发展趋势。本文通过构造用户矩阵与物品矩阵，使用户与物品分别形成不同的特点融合矩阵，一定程度上体现了推荐系统个性化的方式。根据误差函数改进拟牛顿法的步长，将梯度下降模型与拟牛顿法相融合，实现了减小推荐误差和提高算法迭代效率的目标。

下一步，可以在拟合矩阵的基础上融入更多的个性化因子比如标签，浏览记录等，形成用户与物品的个性化肖像。另一方面，从不同的计算过程将梯度下降与拟牛顿法进行融合，例如利用梯度下降改进近似海森矩阵的计算过程，探索减少近似矩阵的存储空间。

参考文献：

[1] Guan Xin, Li Changtun, Guan Yu. Matrix factorization with rating completion: an enhanced SVD model for collaborative filtering recommender systems [J]. IEEE Access, 2017, 5 (99): 27668-27678.

[2] Yuan Ji, Wang Hong, Yi Shangguan, *et al*. Regularized singular value decomposition in news recommendation system [C]//Proc of International Conference on Computer Science & Education. IEEE, 2016.

[3] Hernando A, Ortega F. A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model [J]. Knowledge-Based Systems, 2016, 97(C): 188-202.

[4] Mokhtari A, Ribeiro A. RES: regularized stochastic bfgs algorithm [J]. IEEE Trans on Signal Processing, 2014, 62 (23): 6089-6104.

[5] Badem H, Basturk A, Caliskan A. A new efficient training strategy for deep neural networks by hybridization of rtificial bee colony and limited–memory BFGS optimization algorithms [J]. Neurocomputing, 2017, 44 (15): 22-31.

[6] Köhler S, Wöhner T, Peters R. The impact of consumer preferences on the accuracy of collaborative filtering recommender systems [J]. Electronic Markets, 2017, 26 (4): 1-11.

[7] Elahi M, Ricci F, Rubens N. Active learning in collaborative filtering recommender systems[J]. Computer Science Review, 2016, 20(C): 29-50.

[8] Ghabayen A S, Noah S A M, Ghabayen A S, *et al*. Using tags for measuring the semantic similarity of users to enhance collaborative filtering recommender systems [J]. International Journal on Advanced Science, Engineering and Information Technology, 2017,7(5): 2063-2070.

[9] Morozov S. An empirical study of the recursive input generation algorithm for memory-based collaborative filtering recommender systems [J]. International Journal of Information & Decision Sciences, 2017, 5 (1): 36-49.

[10] Kartoglu I E, Spratling M W. Two collaborative filtering recommender systems based on sparse dictionary coding [J]. Knowledge & Information Systems, 2018 (1): 1-12.

[11] 何洁月，马贝. 利用社交关系的实值条件受限玻尔兹曼机协同过滤推荐算法 [J]. 计算机学报, 2016,39 (1): 183-195. (He Jieyue, Ma Bei. Based on real-valued conditional restricted boltzmann machine and social network for collaborative filtering [J]. Chinese Journal of Computers, 2016,39(1): 183-195.)

[12] 陈婷，朱青，周梦溪,等. 社交网络环境下基于信任的推荐算法 [J]. 软件学报，2017, 28(3) . (Chen Ting, Zhu Qing, Zhou Mengxi,et al. Trust-Based recommendation algorithm in social network [J]. Journal of Software, 2017, 28 (3) .)

[13] 孔欣欣，苏本昌，王宏志，等. 基于标签权重评分的推荐模型及算法研究 [J]. 计算机学报，2017, 40(6): 1440-1452. (Kong Xinxin, Su Benchang, Wang Hongzhi, et al. Research on the modeling and related algorithms of label-weight rating based recommendation system[J]. Chinese Journal of Computers, 2017, 40(6): 1440-1452.)

[14] 徐蕾，杨成，姜春晓,等.协同过滤推荐系统中的用户博弈[J]. 计算机学报,2016,39(6):1176-1189.(Xu Lei, Yang Cheng, Jiang Chunxiao, et al. Game analysis of user participation in collaborative filtering systems[J]. Chinese Journal of Computers, 2016, 39 (6): 1176-1189.)

[15] 黄立威，江碧涛，吕守业，等. 基于深度学习的推荐系统研究综述 [J]. 计算机学报, 2018,41(7):1619-1647.(Huang Liwei, Jiang Bitao, Lv Shouye, et al. Survey on deep learning based recommender systems[J]. Chinese Journal of Computers, 2018,41(7): 1619-1647.)